PROTECTION PROFILE FOR SINGLE-LEVEL OPERATING SYSTEMS IN ENVIRONMENTS REQUIRING MEDIUM ROBUSTNESS

DRAFT 5.31

17 MAY 2000

Table of Contents

List of Figures

List of Tables

1. INTRODUCTION

This section contains overview information necessary to allow a Protection Profile (PP) to be registered through a Protection Profile Registry. The PP identification provides the labeling and descriptive information necessary to identify, catalogue, register, and cross-reference a PP. The PP overview summarizes the profile in narrative form and provides sufficient information for a potential user to determine whether the PP is of interest. The overview can also be used as a stand-alone abstract for PP catalogues and registers. The conventions section provides an explanation of how this document is organized and the glossary of terms section gives a basic definition of terms which are specific to this PP.

1.1 Identification

Title: Protection Profile For Single-Level Operating Systems In Environments Requiring Medium Robustness

Registration: Information Systems Security Organization (ISSO)

Keywords: general-purpose operating system, COTS, medium robustness, single-level, system high, access control, discretionary access control, cryptography

1.2 Overview

This protection profile specifies information security requirements for commercial off-the-shelf (COTS) general-purpose operating systems for use in a single-level medium robustness protected environment. Operating systems meeting these requirements support security assurance and functionality to operate in a medium robustness environment as described in the draft "Guidance and Policy for Department of Defense Information Assurance Memorandum No. 6-8510" (GiG). Threats not countered by the TOE must be addressed by additional security services in the computing environment, in the enclave, and by the enclave boundary protection to provide the required level of robustness.

The intended medium robustness environment may process data at a single level (i.e., System High not to exceed Secret). This capability mandates that operating systems used in this environment must provide Discretionary Access Control (DAC), and cryptographic services. All users will be cleared to the highest level of data in this environment. Access to all data and resources are controlled on the basis of user identity (need-to-know).

The value of information and threat level addressed by this environment establishes the need for medium-grade cryptographic protection of data while it is transmitted within the protected enclave. These cryptographic services can be implemented by hardware, software, or a combination of both. Cryptographic services fully implemented in hardware must meet FIPS 140-1 Level 3 requirements and the augmented requirements identified in this protection profile (PP). For all other implementations, the requirements identified in this PP and FIPS 140-1 Level 1 must be met.

The identified threats establish the need for Evaluated Assurance Level 3 Augmented (EAL3+) and Strength of Mechanism Level 2 (SML-2) as well as the need for additional assurances in the areas of vulnerability analysis/penetration testing, configuration management, development, life-cycle support, and covert channel analysis for cryptography [6].

1.3 Conventions

This document is organized based on Annex B of Part 1 of the Common Criteria (CC). Application notes are integrated with requirements and indicated as notes. These may appear for each component. Application notes document guidance for how the requirement is expected to be applied. Refinements permit additional detail. Refinements are also used to improve the readability of the security functional requirement by substituting generic terms with more specific terminology relevant to the type of TOE or security functionality being described. Extended requirements are requirements not contained in the CC. For additional guidance, the CC itself should be consulted.

1.4 Glossary of Term

This profile uses the terms described in this section to aid in the application of the requirements:

Access is a specific type of interaction between a subject and an object that results in the flow of information from one to the other [7].

An *authorized administrator* is an authorized user who has been granted the authority to manage the TOE. These users are expected to use this authority only in the manner prescribed by the guidance given them.

An *authorized user* is a user who has been properly identified and authenticated. These users are considered to be legitimate users of the TOE.

The *Critical Security Parameters (CSP)* are security-related information (e.g., cryptographic keys, and authentication data such as passwords and PINs, cryptographic seeds) appearing in plaintext or otherwise unprotected form and whose disclosure or modification can compromise the security of a cryptographic module or the security of the information protected by the module [3].

The *Cryptographic Boundary* is an explicitly defined contiguous perimeter that establishes the physical bounds of a cryptographic module [3].

A *Cryptographic key (key)* is a parameter used in conjunction with a cryptographic algorithm that determines [3]:

- the transformation of plaintext data into ciphertext data,
- the transformation of ciphertext data into plaintext data,
- a digital signature computed from data

- the verification of a digital signature computed from data, or
- a data authentication code computed from data

The *Cryptographic Module Security Policy* is a precise specification of the security rules under which a cryptographic module must operate, including the rules derived from the requirements of this standard and additional rules imposed by the vendor [3].

Discretionary Access Control is a means of restricting access to objects based on the identity of subjects and/or groups to which they belong. The controls are discretionary in the sense that a subject with a certain access permission is capable of passing that permission (perhaps indirectly) on to any other subject (unless restrained by mandatory access control) [7].

An *Enclave* is an environment that is under the control of a single authority and has a homogeneous security policy, including personnel and physical security and therefore protected from other environments. Local and remote elements that access resources within an enclave must satisfy the policy of the enclave. Enclaves can be specific to an organization or a mission and may also contain multiple networks. They may be logical, such as an operational area network (OAN) or be based on physical location and proximity [2].

The *Level of Robustness* is the characterization of the strength of a security function, mechanisms, service or solution, and the assurance (or confidence) that it is implemented and functioning correctly. DoD has three levels of robustness [2]:

a. *High*: Security services and mechanisms that provide the most stringent available protection and rigorous security countermeasures

b. *Medium*: Security services and mechanisms that provide for layering of additional safeguards above the DoD minimum.

c. *Basic*: Security services and mechanisms that equate to good commercial practices.

Mission Category reflects the importance of information relative to the achievement of DoD goals and objectives, particularly the warfighter’s combat mission. Mission categories are primarily used to determine requirements for availability and integrity services. DoD has three mission categories [2]:

a. *Mission Critical*. Systems handling information which is determined to be vital to the operational readiness or mission effectiveness of deployed and contingency forces in terms of both content and timeliness and must be absolutely accurate and available on demand (may include classified information in a traditional context, as well as sensitive and unclassified information).

b. *Mission Support*. Systems handling information that is important to the support of deployed and contingency forces; must be absolutely accurate, but can sustain minimal delay without seriously affecting operational readiness or mission effectiveness (may be classified information, but is more likely to be sensitive or unclassified information).

c. *Administrative*. Systems handling information which is necessary for the conduct of day-to- day business, but does not materially affect support to deployed or contingency forces in the short term (may be classified information, but is much more likely to be sensitive or unclassified information).

An *Operating Environment* is the total environment in which an information system operates. It includes the physical facility and controls, procedural and administrative controls, and personnel controls (e.g., clearance level of the least cleared user) [2].

The *Target of Evaluation (TOE)* is an IT product or system and its associated administrator and user guidance documentation that is the subject of an evaluation [1].

The *TOE Security Functions (TSF)* is a set consisting of all hardware, software, and firmware of the TOE that must be relied upon for the correct enforcement of the TSP [1].

1.5 Document Organization

Section 2 of this document describes the Target of Evaluation in terms of its envisaged usage and connectivity.

Section 3 defines its security environment in terms of the threats to its security, the security assumptions made about its use, and the security policies that must be followed.

Section 4 identifies the security objectives derived from these threats and policies.

Section 5 identifies and defines the security functional requirements from the Common Criteria that must be met by the TOE in order for the functionality-based objectives to be met.

Section 6 identifies the security assurance requirements.

Section 7 presents the rationale that justifies the security requirements included herein, thereby linking Sections 3, 4, 5, and 6.

An acronym list is provided to define frequently used acronyms.

A reference section is provided to identify background material.

2. Target of Evaluation (TOE) Description

This protection profile specifies requirements for general-purpose multi-user operating systems. Such systems are typically employed in an office automation environment and contain file systems, printing services, network services and data archival services and can host other applications (e.g., mail, databases).

Operating systems meeting these requirements can operate in a protected enclave (as shown in Figure 2.1) that is accredited to connect to untrusted but controlled networks[1]. Such an enclave will consist of clients and servers with numerous applications available to users. In this environment, the TOE operating systems may be accessible by external IT systems that are beyond the security policies of the enclave. The users of these IT systems are similarly beyond the control of the operating system's policies. Although the users of these controlled systems are trusted to a certain extent, they are outside the scope of control of this particular enclave so nothing can be presumed about their intent, so they must be viewed as hostile.

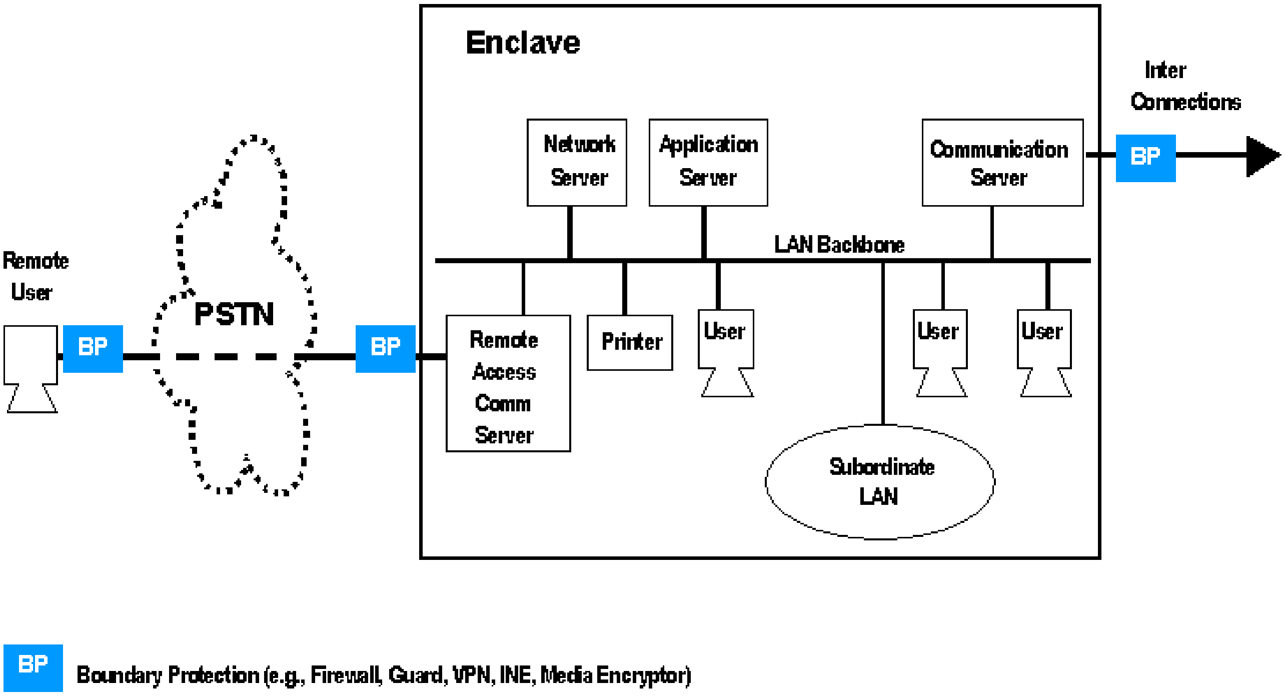

Figure 2.1 - Protected Enclave

Access to all data and resources protected by compliant TOEs are controlled on the basis of user identity (need-to-know). Operating systems meeting these requirements provide sufficient assurance and functionality to contribute to data protection in the medium robustness

[1] Interconnections of DoD systems to uncontrolled networks shall be accomplished by processes consistent with the philosophy of the Secret and Below Interoperability (SABI) process [2].

environment [2]. This level of robustness is provided by the combinations of physical security, security services, infrastructures, and processes provided by all the layers in the protected environment. This enclave is assumed to be under the control of a single authority and has a homogeneous security policy. This profile does not specify any security characteristics of security hardened devices (e.g. guards, firewalls) that protect the enclave at its external boundary. However, it is presumed that such boundary protection is provided. The information value of this environment is such that violation of the information protection policy would cause serious damage to the security, safety, financial posture, and/or infrastructure of the organization [6]. The threat level addressed by this environment is such to counter sophisticated adversary with moderate resources who is willing to take little risk (e.g., organized crime, sophisticated hackers) [6].

The information processed by such systems can be mission critical. Specific security features required for these systems include:

- Identification and Authentication which mandates authorized users identify and authenticate before accessing information stored on the system;
- Discretionary Access Control (DAC) which allows authorized users to specify protection for data files that they create;
- Cryptographic services to allow authorized users and applications to encrypt and digitally sign data as it resides within the system and as it is transmitted to other systems; and
- Audit services to allow system administrators to detect and analyze potential security violations.

Compliant TOEs do not provide mechanisms to ensure availability of data residing on the TOE or services provided by the TOE. Nor do they provide any physical protection mechanisms. Such protection must be provided by the environment (e.g., through mirrored/duplicated data).

3. TOE Security Environment

The security environment for the functions addressed by this specification includes threats, security policy, and usage assumptions and security objectives, as discussed below.

3.1 Threats

Specific threats to IT security that should be countered by the operating system:

T.AUDIT_CORRUPT	Audit data may be tampered with or lost.
T.CONFIG_CORRUPT	Configuration data or other trusted data may be tampered with or lost.
T.DOS	Denial of service attacks may occur.
T.EAVESDROP	Unauthorized access by either an insider or outsider of data in transit sent or received by the IT operating system.
T.IMPROPER_ADMIN	Improper administration resulting in defeat of specific security features.
T.MASQUERADE	An entity on one machine on the network may masquerade itself as an entity on another machine on the same network.
T.OBJECTS_NOT_CLEAN	Systems may not adequately remove the data from objects between uses by different users, thereby releasing information to the subsequent user
T.POOR_DESIGN	Unintentional or intentional errors in requirements specification, design or development of the IT operating system.
T.POOR_IMPLEMENTATION	Unintentional or intentional errors in implementing the design of the IT operating system.
T.POOR_TEST	Incorrect system behavior resulting from inability to demonstrate that all functions and interactions within the operating system operation are correct.
T.REPLAY	An attacker may gain access by replaying authentication (or other) information.
T.SPOOF	Authorized users incorrectly believe they are communicating with the IT operating system, when they are in fact communicating with a hostile entity masquerading as the IT operating system.
T.SYSACC	An attacker may gain unauthorized access to the administrator account, or that of other trusted personnel.
T.UNAUTH_ACCESS	Unauthorized access by either an insider or outsider of data

on the IT system.

T.UNAUTH_MODIFICATION Unauthorized modification or use of IT operating system attributes and resources.

T.UNDETECTED_ACTIONS Failure to identify and record unauthorized actions.

T.UNSECURE_START Upon failure of the IT operating system, reboot results in insecure state of the operating system.

T.USER_CORRUPT User data may be lost or tampered with by other users.

3.2 Security Policy

Policy statements whose enforcement must be provided by the operating system’s security mechanisms:

P.ACCOUNT The users of the system shall be held accountable for their actions within the system.

P.AUTHORIZATION The system must have the ability to limit the extent of each user's authorizations.

P.AUTHORIZED_USERS Only those users who have been authorized to access the information within the system may access the system.

P.IANDA All users will be identified and authenticated prior to accessing any controlled resources.

P.NEED_TO_KNOW The system must limit the access to the information in protected resources to those authorized users who have a need to know that information.

P.REMOTE_ADMIN_ACCESS System Administrators may access their workstations remotely (i.e., from outside the domain).

P.ROLES The authorized administrator and cryptographic administrator will have separate and distinct roles associated with them

P.SELF_PROTECTION The operating system shall maintain a domain for its own execution that protects it from external interference or tampering.

P.SYSTEM_INTEGRITY The system must have the ability to periodically validate its correct operation and, with the help of administrators, it must be able to recover from any errors that are detected.

P.TESTING The operating system will undergo independent testing as part of an independent vulnerability analysis.

P.TRACE The operating system must have the ability to review the actions of individuals.

P.TRUSTED_RECOVERY Procedures and/or mechanisms shall be provided to assure that, after a system failure or other discontinuity, recovery without a protection compromise is obtained

P.VULNERABILITY_SEARCH The system will undergo an analysis for vulnerabilities beyond those that are obvious.

3.3 Security Usage Assumptions

Assumptions about the use of the IT operating system:

A.ADMIN Administrators are competent, well trained, and trustworthy.

A.ADMIN_ACCESS Administrator administration will be performed access their workstations directly (i.e., not remotely).

A.MANAGE There will be at least one competent individual assigned to manage the TOE and the security of the information it contains.

A.PHYSICAL It is assumed that physical security will be provided within the domain appropriate for the value of the IT assets protected by the operating system and the value of the stored, processed, and transmitted information.

4. Security Objectives

Security objectives to be met by the IT operating system:

O.ACCESS The IT operating system must ensure that only authorized users gain access to it and to its resources that it controls.

O.AUDIT_GENERATION The IT operating system must provide the capability to detect audit events associated with individual users and create records thereof.

O. AUDIT_PROTECTION The IT operating system must provide the capability to protect audit information associated with individual users.

O. AUDIT_REVIEW The IT operating system must provide the capability to selectively view audit information associated with individual users.

O.CONFIG_MGMT All changes to the operating system and its development evidence will be tracked and controlled.

O.DISCRETIONARY_ACCESS The IT operating system must control accesses to resources based upon the identity of users or groups of users. The IT operating system must allow authorized users to specify which resources may be accessed by which users or groups of users.

O.ENCRYPTED_CHANNEL Encryption will be used to provide confidentiality protection of user data in transit; the encryption is assumed to be provided by application hardware or software.

O.INSTALL The IT operating system must be delivered, installed, managed, and operated in a manner that maintains IT security.

O.MANAGE The IT operating system must provide all the functions and facilities necessary to support the authorized administrators in their management of the security of the IT system.

O.PENETRATION_TEST Evidence must be provided to show adequate system design and test.

O.PROTECT The IT operating system must provide means to protect its own data and resources.

O.RECOVERY Procedures and/or mechanisms shall be provided to assure that, after a system failure or other discontinuity, recovery without a protection compromise is obtained

O.RESIDUAL_INFORMATION The IT operating system must ensure that any information contained in a protected resource is not released when the resource is recycled.

O.SELF_PROTECTION The operating system shall maintain a domain for its own execution that protects it from external interference or tampering.

O.SOUND_DESIGN The design of the IT operating system is the result of sound design principles, with the development documentation found to be adequate.

O.SOUND_IMPLEMENTATION The implementation of the IT operating system is an accurate instantiation of its deign.

O.TESTING The operating system will undergo independent testing, based at least in part upon an independent vulnerability analysis.

O.TRAINED_ ADMIN System Administrators and all others with authorized access to the IT system shall be adequately trained.

O.TRAINED_USERS Users with authorized access to the IT operating system shall be adequately trained.

O.TRUSTED_PATH The operating system will provide a means to ensure users are not communicating with some other entity pretending to be the operating system.

O.USER_IDENTIFICATION The operating system shall identify individuals.

O.VULNERABILITY_ANALYSIS The system will undergo an analysis for vulnerabilities beyond those that are obvious.

5. Security Functional Requirements

This section contains the detailed security functional requirements for the trusted security functions (TSF) of operating systems supporting single-level secret and below systems in an environment requiring medium robustness. These security functional requirements are selected from Part 2 of the Common Criteria.

5.1 Security Audit

5.1.1 Security Audit Automatic Response

FAU_ARP.1.1 The TSF shall generate a warning for the authorized administrator upon detection of a potential security violation.

5.1.2 Security Audit Data Generation

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shutdown of the audit functions;

b) All auditable events listed in Table 1, below;

c) Start-up and shutdown of the operating system;

d) Uses of special permissions (e.g., by administrators) that circumvent the access control policies;

e) Other auditable events defined in the ST.

Requirement	**Audit events prompted by requirement**
FAU_ARP.1	• Actions taken due to imminent security violations
FAU_GEN.1	(none)
FAU_GEN.2	(none)
FAU_SAA.1	• Enabling and disabling of any of the analysis mechanisms. • Automated responses provided by the tool.
FAU_SAR.1	• Reading of information from the audit records.
FAU_SAR.2	• Unsuccessful attempts to read information from the audit records
FAU_SAR.3	(none)
FAU_SEL.1	• All modifications to the audit configuration that occur while the audit collection functions are operating.
FAU_STG.1	(none)

FAU_STG.4	• Actions taken due to the audit storage failure.
FCS_CKM.1	• Success and failure of the activity. • The object attribute(s), and object value(s) excluding any sensitive information (e.g. secret or private keys).
FCS_CKM.2	• Success and failure of the activity. • The object attribute(s), and object value(s) excluding any sensitive information (e.g. secret or private keys).
FCS_CKM.3	• Success and failure of the activity. • The object attribute(s), and object value(s) excluding any sensitive information (e.g. secret or private keys).
FCS_CKM.4	• Success and failure of the activity. • The object attribute(s), and object value(s) excluding any sensitive information (e.g. secret or private keys).
FCS_COP.1	• Success and failure, and the type of cryptographic operation. • Any applicable cryptographic model(s) of operation, subject attributes and object attributes.
FDP_ACC.2	(none)
FDP_ACF.1	• All requests to perform an operation on an object covered by the SFP.
FDP_ETC.1	• All attempts to export user data, including any security attributes.
FDP_ETC.2	• All attempts to export information.
FDP_IFC.2	(none)
FDP_IFF.2:	• All decisions on requests for information flow.
FDP_IFF.3	• All decisions on requests for information flow. • The use of identified illicit information flow channels.
EXTENDED_FDP_INC.2	(none)
EXTENDED_FDP_INF.2:	• All decisions on requests for information flow.
FDP_ITC.1	• All attempts to import user data, including any security attributes.
FDP_ITC.2	• All attempts to import user data, including any security attributes.
FDP_ITT.1	• All attempts to transfer of user data, including identification of the protection method used and any error that occurred.
FDP_RIP.2	(none)

EXTENDED_FDP_RIP.2	(none)
FIA_AFL.1	• The reaching of the threshold for the unsuccessful authentication attempts and the actions (e.g. disabling of a terminal) taken and the subsequent, if appropriate, restoration to the normal state (e.g. re-enabling of a terminal).
FIA_ATD.1	(none)
FIA_SOS.1	• Rejection or acceptance by the TSF of any tested secret.
FIA_UAU.2	• All use of the authentication mechanism
FIA_UAU.7	(none)
FIA_UID.2	• All use of the user identification mechanism, including the user identity provided.
FIA_USB.1	• Success and failure of binding of user security attributes to a subject (e.g. success and failure to create of a subject).
FMT_MOF.1	• All modifications in the behavior of the functions in the TSF.
FMT_MSA.1	• All modifications of the values of security attributes.
FMT_MSA.2	• All offered and rejected values for a security attribute.
FMT_MSA.3	• Modifications of the default setting of permissive or restrictive rules. • All modifications of the initial values of security attributes.
FMT_MTD.1	• All modifications of the values of TSF data, including audit data.
FMT_REV.1	• All attempts to revoke security attributes.
FMT_SAE.1	• Specification of the expiration time for an attribute • Action taken due to attribute expiration.
FMT_SMR.1	• Modifications to the group of users that are part of a role.
FMT_SMR.3	• Explicit requests to assume a role. • Use of any function restricted to an administrator role (identified in FMT_SMR.1).
FPT_AMT.1	• Execution of the tests of the underlying machine and the results of the tests.
FPT_ITT.1	(none)
FPT_ITT.3	• Detection of modification of TSF data
FPT_RCV.1	• The fact that a failure or service discontinuity occurred. • Resumption of the regular operation.

	• Type of failure or service discontinuity
FPT_RVM.1	(none)
FPT_SEP.1	(none)
FPT_STM.1	• Changes to the time.
FPT_TDC.1	• Successful use of TSF data consistency mechanisms. • Use of TSF data consistency mechanisms.
FPT_TRC.1	• Restoring consistency upon reconnection. • Detected inconsistency between TSF data.
FPT_TST.1	• Execution of the TSF self tests and the results of the tests.
EXTENDED_FPT_CTST.1	• Execution of the TSF self tests and the results of the tests.
FRU_RSA.1	• All attempted uses of the resource allocation functions for resources that are under control of the TSF.
FTA_SSL.1	• Locking of an interactive session by the session locking mechanism. • Any attempts at unlocking of an interactive session.
FTA_SSL.2	• Locking of an interactive session by the session locking mechanism. • Any attempts at unlocking of an interactive session.
FTA_TAB.1	(none)
FTA_TAH.1	(none)
FTP_TRP.1	• All attempted uses of the trusted path functions. • Identification of the user associated with all trusted path failures, if available
EXTENDED_FTP_TRP.1	• All attempted uses of the trusted path functions. • Identification of the user associated with all trusted path failures, if available.

Table 5.1 - Auditable Events

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST:

- the identity of the object;

- the sensitivity and integrity labels of the subject;
- for changes to TSF data, the new value (except authentication data and cleartext cryptographic variables, such as key variables, seed, etc.);
- for authentication attempts, the origin of the attempt (e.g., terminal identifier);
- for uses of a role, the type of role, and the origin of its request;
- other audit relevant information identified in the ST.

FAU_GEN.2.1 The TSF shall be able to associate each auditable event with the identity of the user that caused the event.

Application Note: For failed login attempts no user association is required because the user is not under TSF control until after a successful identification/authentication.

5.1.3 Security Audit Analysis

FAU_SAA.1.1 The TSF shall be able to apply a set of rules in monitoring the audited events and based upon these rules indicate a potential violation of the TSP.

FAU_SAA.1.2 The TSF shall enforce the following rules for monitoring audited events:

a) Accumulation or combination of auditable events (as defined in the ST) known to indicate a potential security violation;

b) any other rules identified in the ST.

5.1.4 Security Audit Review

FAU_SAR.1.1 The TSF shall provide authorized administrators with the capability to read all audit information from the audit records.

FAU_SAR.1.2 The TSF shall provide the audit records in a manner suitable for the user to interpret the information.

Application Note: The intent of this requirement is that there be a tool for authorized administrators to access the audit trail. The manner in which this is provided is an implementation detail, but its implementation must allow the administrator to make effective use of the presented information. It is expected (yet not necessary) that the tool satisfying this requirement will also satisfy the FAU_SAR.3 and FAU_SEL.1 requirements.

FAU_SAR.2.1 The TSF shall prohibit all users read access to the audit records, except those users that have been granted explicit read-access.

FAU_SAR.3.1 The TSF shall provide the ability to perform searches and sorting of audit data based on the following attributes:

a) User identity;

b) Date of the event;

c) Time of the event;

d) Type of event;

e) any additional attributes identified by the ST author.

5.1.5 Security Audit Event Selection

FAU_SEL.1.1 The TSF shall be able to include or exclude auditable events from the set of audited events based on the following attributes:

a) object identity;

b) user identity;

c) host identity;

d) event type

e) any additional attributes identified by the ST author.

5.1.6 Security Audit Event Storage

FAU_STG.1.1 The TSF shall protect the stored audit records from unauthorized deletion.

FAU_STG.1.2 The TSF shall be able to prevent modifications to the audit records.

Application Note: In order to reduce the performance impact of audit generation, audit records are often temporarily buffered in memory before being written to the disk. In such implementations, these buffered records will be lost if the operation of the TOE is interrupted by hardware or power failures. The developer must document the expected loss in such circumstances and show that it has been minimized.

FAU_STG.4.1 - Refinement: When the audit trail becomes full, the TSF shall generate an alarm to the authorized administrator and generally prevent auditable events (except those generated by the authorized administrator in the context of performing TOE maintenance).

Application Note: The requirement of "preventing" auditable events upon storage resource exhaustion is a minimum functionality; providing a range of configurable choices (e.g., ignoring auditable actions and/or changing to a degraded mode) is allowable, provided that "preventing" auditable events is among the choices. (If choices are configurable, then FMT_MOF.1 should be included in the ST.)

5.2 Cryptographic Support

5.2.1 Cryptographic Key Management

5.2.1.1 Cryptographic Key Generation

FCS_CKM.1.1 - Refinement: The TSF shall generate cryptographic keys in accordance with random number generators (RNGs)/pseudorandom number generators (PRNGs) and all self-tests as specified in this PP.

Refinement: The key generation process shall include two or more RNG/PRNG components (e.g., two or more RNGs/PRNGs, NIST-approved algorithms, or combinations thereof) such that each component is independently in accordance with RNGs/PRNGs as specified in this PP.

Application Note: If a NIST-approved algorithm is used as a component of the key generation process, the Advanced Encryption Standard (AES) employing key lengths of 128 bits or greater will be required. However, since AES is not expected to be fully established for at least 18 months, in the interim the Triple Data Encryption Algorithm (TDEA) employing three keys (i.e., 168 bits keys) that meets the following: FIPS PUB 140-1, Level 3, FIPS PUB 46-3, and ANSI X9.52 will be approved in place of AES.

EXTENDED_FCS_CKM.1.2: The TSF shall append parity bits or checkwords to each generated key.

EXTENDED_FCS_CKM.1.3: Generation of public key certificates shall be in accordance with DoD PKI Class 4[2] X.509, version 3 certificate requirements.

5.2.1.2 Cryptographic Key Distribution

FCS_CKM.2.1 - Refinement: If keys are not generated and retained at the user's cryptomodule, then the following key distribution methods shall be employed:

a) Physical Methods

- The TSF shall physically distribute traditional key in accordance with a specified cryptographic key distribution method as identified in FIPS PUB 140-1.
- The TSF shall physically distribute public key material (certificates and/or keys) in accordance with the DoD PKI for public key distribution using Class 4[2] X.509, version 3 certificates with hardware tokens for protection of private key used by the System and Cryptosecurity Administrators that meet the following: DoD PKI Roadmap, DoD X.509 Certificate Policy, and FIPS PUB 171--Key Management Using ANSI X9.17.

b) Electronic Methods

- The TSF shall electronically distribute public key material (certificates and/or keys) in accordance with the DoD PKI for public key distribution using Class 4 X.509, version 3 certificates that meet the following: DoD PKI Roadmap, DoD X.509 Certificate Policy, and FIPS PUB 171--Key Management Using ANSI X9.17.

[2] DoD System High (not to exceed Secret) applications require Class 5 PKI, but currently this class is just a concept. In the interim, Class 4 X.509, version 3 certificate requirements are approved under the added requirement that stronger protection mechanisms must be applied at the boundaries of the protected environment as stated earlier in this PP. When Class 5 certificates are fully established, they will be required.

5.2.1.3 Cryptographic Key Access

FCS_CKM.3.1 - Refinement: Any TSF encryption key archiving shall be performed in accordance with FIPS PUB 140-1. Signature keys shall not be archived.

5.2.1.4 Cryptographic Key Destruction

FCS_CKM.4.1: The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method (immediate and complete zeroization of all plaintext cryptographic keys and all other critical security parameters within the device) that meets the following: FIPS PUB 140-1, Level 4.

Refinement: Upon each issuance of the zeroization, the destruction shall be executed by overwriting the key/critical security parameter storage area three or more times with an alternating pattern.

EXTENDED_FCS_CKM.4.2: Each intermediate storage area for key/critical security parameter shall be immediately and completely zeroized upon the key's transfer to another location.

5.2.1.5 Internal Cryptographic Key Handling and Storage

EXTENDED_FCS_CKM.5.1: The TSF shall perform key entry and output in accordance with FIPS PUB 140-1, Level 4.

EXTENDED_FCS_CKM.5.2: The TSF shall provide a means to ensure that keys are associated with the correct entities (i.e., person, group, or process) to which the keys are assigned.

EXTENDED_FCS_CKM.5.3: The TSF shall perform a key error detection check on each transfer of key (internal, intermediate transfers).

EXTENDED_FCS_CKM.5.4: The TSF shall encrypt or split secret and private keys when not in use.

5.2.2 Cryptographic Operation

FCS_COP.1.1 - Refinement: The TSF shall perform data encryption/decryption services in accordance with a specified cryptographic algorithm *(pending)* NIST-approved Advanced Encryption Standard (AES) which incorporates key lengths of 128 bits or greater, and meets the *(pending)* AES FIPS PUB, other existing standards deemed applicable to AES (as below for TDEA*), and all other requirements in this PP.*

Application Note: Since AES is not expected to be fully established for at least 18 months, in the interim the TSF shall perform data encryption/decryption services in accordance with the TDEA cryptographic algorithm employing three keys (i.e., 168 bit key) that meets the following: FIPS PUB 140-1, Level 3, FIPS PUB 46-3, ANSI X9.52, and any augmented requirements identified in this PP.

FCS_COP.1.1 - Refinement: The TSF shall perform cryptographic signature services in accordance with the NIST-approved Digital Signature Algorithm (DSA) and a modulus of 3000 bits or greater, or in accordance with a NIST-approved Elliptic Curve Digital Signature Algorithm (ECDSA) with a corresponding key size. These services must meet FIPS PUB 186-2.

FCS_COP.1.1 - Refinement: The TSF shall perform cryptographic hashing services in accordance with the NIST-approved Secure Hash Algorithm (SHA-1) employing a 160-bit message digest. These services must meet FIPS PUB 180-1.

Application Note: Future migration to incorporate stronger cryptographic hashing services (i.e., with a digest corresponding to double the system encryption key strength) will be required when such NIST standards are established.

FCS_COP.1.1 - Refinement: The TSF shall perform cryptographic key exchange services in accordance with the Diffie-Hellman Algorithm and cryptographic key size of 3000 bits or greater, or in accordance with a NIST-approved Elliptic Curve Key Exchange Algorithm (ECKEA) that meets the following: ANSI X 9.42--Agreement of Symmetric Keys Using Discrete Logarithm Cryptography.

FCS_COP.1.1 - Refinement: The TSF shall perform random number generation (RNG)/ pseudorandom number generation (PRNG) services in accordance with all the RNG/PRNG self-tests of FIPS PUB 140-1, Level 4 and the augmented self-test requirements of this PP.

5.3 User Data Protection

5.3.1 Access Control Policy

FDP_ACC.2.1 The TSF shall enforce the Discretionary Access Control policy on all subjects and objects and operations among subjects and objects covered by the policy.

Application Note: The DAC policy need not cover public objects (i.e., objects that, because of their nature are readable by all users yet writable by none (e.g. the system clock), or vice versa.

FDP_ACC.2.2 The TSF shall ensure that all operations between any subject in the TSC and any object within the TSC are covered by an access control policy.

5.3.2 Access Control Functions

FDP_ACF.1.1 The TSF shall enforce the Discretionary Access Control policy to objects based on the following:

a) The user identity and group membership(s) associated with a subject; and

b) The access control attributes associated with an object with:

- the ability to associate allowed or denied operations with one or more user identities;
- the ability to associate allowed or denied operations with one or more group identities; and
- defaults for allowed or denied operations.

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

a) For each operation there shall be a rule, or rules, that use the permission attributes where the user identity of the subject matches a user identity specified in the access control attributes of the object;

b) For each operation there shall be a rule, or rules, that use the permission attributes where the group membership of the subject matches a group identity specified in the access control attributes of the object; and

c) For each operation there shall be a rule, or rules, that use the default permission attributes specified in the access control attributes of the object when neither a user identity or group identity matches.

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rule:

a) Authorized Administrators must follow the above-stated Discretionary Access Control policy, except in special circumstances (such as invoking a privilege), as described by the ST author.

b) Other rules identified by the ST author.

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the rules:

- as identified by the ST author.

Application Note: Because the rules that govern the DAC policy may be implementation-specific, they must be specified in the ST. In completing the rule assignment above, the resulting mechanism must be able to specify access rules that are user-specific. (The user in such a rule may have special status, e.g., owner of the object.) The mechanism must also support specifying access to the membership of individual groups. Conformant implementations include self/group/world permission bits and access control lists.

If the TOE has public objects, there must be rules for accessing.

Exceptions to the basic policy for access by authorized administrators or other forms of special authorization; must be covered under FDP_ACF.1.3.

The ST must list the attributes that are used by the DAC policy for access decisions, such as permission bits, access control lists, and object ownership. A single set of attributes may be associated with multiple objects, such as all objects stored on a single floppy disk. The association may also be indirectly bound to the object, such as when the attributes are associated with the name of the object, rather than with the object itself.

5.3.3 Export to Outside TSF Control

FDP_ETC.1.1 The TSF shall enforce the Discretionary Access Control policy when exporting user data, controlled under the SFPs, outside of the TSC.

FDP_ETC.1.2 The TSF shall export the user data without the user data's associated security attributes.

5.3.4 Import From Outside TSF Control

FDP_ITC.1.1 The TSF shall enforce the Discretionary Access Control policy when importing unlabeled user data, controlled under the SFP, from outside the TSC.

Application Note: The "label" is the security attributes associated with the data. In the context of the Discretionary Access Control policy, this "label" is the ACL, permission bits, or whatever other identity-based security attributes are associated with the objects.

FDP_ITC.1.2 The TSF shall ignore any security attributes associated with the unlabeled user data when imported from outside the TSC.

FDP_ITC.1.3 The TSF shall enforce the following rules when importing unlabeled user data controlled under the SFP from outside the TSC:

a) When importing data that has no associated DAC attributes, the data is to be given the DAC attributes of the importer of the data.

b) (any additional importation control rules identified in the ST).

Application Note: The ST author must explicitly state the rules under which authorized users can designate the security attributes of the mechanisms, or devices, used to import data without security attributes; and any attribute change must be audited. The ST author must also make it clear that mechanisms, or devices, used to import data without security attributes cannot also be used to import data with security attributes unless this change in state can only be done manually and is audited.

5.3.5 Internal TOE Transfer

FDP_ITT.1.1 The TSF shall enforce the Discretionary Access Control policy to prevent the disclosure and modification of user data when it is transmitted between physically-separated parts of the TOE.

Application Note: If distributed, the TOE must provide a mechanism to protect data transmitted from one part of the TOE to another. This mechanism, whether link encryption, application-level protection (SHTTP), or some other mechanism, must be described in the ST.

5.3.6 Residual Information Protection

FDP_RIP.2.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the allocation or deallocation of that resource to all objects.

Application Note: This requirement applies to all resources governed by or used by the TSF; it includes resources used to store data and attributes. It also includes the encrypted representation of information.

Clearing the content of resources on deallocation is sufficient to satisfy this requirement, provided that unallocated resources will not accumulate new information until they are allocated again.

5.3.7 Extension: Residual Cryptographic Key Information Protection

EXTENDED_FDP_RIP.2.1 The TSF shall ensure that any resource containing critical security parameters is cleared of all information upon the deallocation of that resource by overwriting its contents as defined for key destruction in this PP.

Application Note: The data area for critical security parameters is kept isolated from other data (see FPT_SEP.1).

5.4 Identification and Authentication

5.4.1 Authentication Failures

FIA_AFL.1.1 The TSF shall detect when an administrator-specified number of unsuccessful authentication attempts occur related to any user login process performed under its direct control.

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall close the connection (reinitiate the login process) for that login session except for system administrator sessions.

5.4.2 User Attribute Definition

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users:

a) User Identifier;

b) Group Memberships;

c) Authentication Data;

d) Security-relevant Roles; and

e) Any other security attributes identified in the ST.

Application Note: The specified attributes are those that are required by the TSF to enforce the DAC policy, the generation of audit records, and proper identification and authentication of users. The user identity must be uniquely associated with a single individual user.

Group membership may be expressed in a number of ways: a list per user specifying to which groups the user belongs, a list per group which includes which users are members, or implicit association between certain user identities and certain groups.

A TOE may have two forms of user and group identities: a text form and a numeric form. In these cases there must be unique mapping between the representations.

It is possible that the notion of privilege is tied to the security-relevant roles (item d).

5.4.3 Specification of Secrets

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet the following:

a) For each attempt to use the authentication mechanism, the probability that a random attempt will succeed is less than one in 250,000,000,000,000;

Application Note: This can be achieved with a password of eight characters, assuming an alphabet of 60 characters.

b) The authentication mechanism must provide a delay between attempts, such that there can be no more than ten attempts per minute; and

c) Any feedback given during an attempt to use the authentication mechanism will not reduce the probability below the above metrics.

Application Note: The ST must specify the method of authentication. Where authentication is provided by a password mechanism, the ST must show that the restrictions upon passwords (length, alphabet, and other characteristics) result in a password space conforming to item (a) above, as well as characterize the delay to show conformance to item (b) above. Where authentication is provided by a mechanism other than passwords, the ST must show authentication method has a low probability that authentication data can be forged or guessed.

5.4.4 User Authentication

FIA_UAU.2.1 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

Application Note: The ST must specify the (non-TSF-mediated) actions that are allowed to an unauthenticated user.

FIA_UAU.7.1 The TSF shall provide only obscured feedback to the user while the authentication is in progress.

Application Note: "Obscured feedback" implies the TSF does not produce a visible display of any authentication data entered by a user (such as the echoing of a password), although an obscured indication of progress may be provided (such as an asterisk for each character).

5.4.5 User Identification

FIA_UID.2.1 The TSF shall require each user to identify itself before allowing any other TSF-mediated actions on behalf of that user.

Application Note: The ST must specify the (non-TSF-mediated) actions that are allowed to an unidentified user.

5.4.6 User-Subject Binding

FIA_USB.1.1 The TSF shall associate the appropriate user security attributes with subjects acting on behalf of that user.

Application Note: The term " appropriate" applies as follows:

- *The user identity which is associated with auditable events;*
- *The user identity or identities which are used to enforce the Discretionary Access Control Policy;*
- *The group membership or memberships used to enforce the Discretionary Access Control Policy;*
- *The certificate used to represent the user;*
- *The key used to encrypt data on behalf of the user;*
- *Other attributes identified in the ST.*

Application Note: The DAC policy and audit generations require that each subject acting on behalf of a user has a user identity associated with the subject. While this identity is typically the one used at the time of identification to the system, the DAC policy enforced by the TSF may include provisions for making access decisions based upon a different user identity, such as the "set user ID (su)" command in UNIX. For TOEs with such a capability, the ST must include a description of how the TSF maintains the association between the new user ID and the user.

Application Note: The attributes listed in FIA_USB.1 should be comparable to those listed in FIA_ATD.1.

5.5 Security Management

5.5.1 Management of Security Functions Behavior (audit generation functions)

FMT_MOF.1.1 The TSF shall restrict the ability to determine the behavior of, disable, enable, and modify the behavior of the functions related to audit generation to the authorized administrators.

Application Note: The "functions related to audit generation" applies only to the selection of which events are to be audited.

While the TOE may provide audit generation functions in addition to those listed above, these additional functions must be similarly restricted to the authorized administrator.

5.5.2 Management of Security Functions Behavior (authentication data-changing functions)

FMT_MOF.1.1 The TSF shall restrict the ability to enable the functions associated with changing the values of user authentication data to authorized administrators and users authorized to modify their own authentication data.

Application Note: This component applies only to security functions used to change a user password, or whatever other authentication data is used.

While the TOE may provide authentication data-changing functions in addition to those listed above, these additional functions must be similarly restricted to the authorized administrator.

5.5.3 Management of Security Attributes

FMT_MSA.1.1 The TSF shall enforce the Discretionary Access Control policy to restrict the ability to query or modify the security attributes to authorized administrators and users authorized by the Discretionary Access Control Policy to modify object security attributes.

Refinement: The ST must state the components of the access rights that may be modified, and must state any restrictions that may exist for each type of authorized user (i.e., the components of the access rights that the user is allowed to modify). The ability to modify access rights must be restricted such that a user having access rights to a named object does not have the ability to modify those access rights unless granted the right to do so. This restriction may be explicit, based upon the object ownership, or based upon a set of object hierarchy rules.

Application Note: "Modify" is understood to include deleting and setting attributes.

FMT_MSA.2.1 The TSF shall ensure that only secure values are accepted for security attributes.

Application Note: The identity attributes are listed in FDP_ACF.1, FDP_IFC.1, and FIA_ATD.1.

FMT_MSA.3.1 The TSF shall enforce the Discretionary Access Control policy to provide restrictive default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the authorized administrator to specify alternative initial values to override the default values when an object or information is created.

Application Note: The TOE must provide protection by default for all objects at creation time. This may be accomplished through the enforcement of a restrictive default access on objects, or through requiring the user to explicitly specify the desired access controls upon the object at its creation, provided that *there is no window of vulnerability through which unauthorized access may be gained to newly-created objects.*

5.5.4 Management of TSF Data (general)

FMT_MTD.1.1 The TSF shall restrict the ability to create, query, modify, delete, or clear the security-relevant TSF data to the authorized administrator.

Application Note: This component applies only to security attributes that are used to maintain the TSP.

While the TOE may provide TSF data management actions in addition to those listed above, these additional actions must be similarly restricted to the authorized administrator.

5.5.5 Management of TSF Data (audit records)

FMT_MTD.1.1 The TSF shall restrict the ability to create, delete or clear the audit records to authorized administrators.

Application Note: This selection of "create, delete, or clear" functions for audit trail management reflect common management functions.

While the TOE may provide TSF data management actions in addition to those listed above, these additional actions must be similarly restricted to the authorized administrator.

5.5.6 Management of TSF Data (user security attributes)

FMT_MTD.1.1 The TSF shall restrict the ability to initialize and modify the user security attributes, other than authentication data, to authorized administrators.

Application Note: This component applies only to security attributes that are used to maintain the TSP.

While the TOE may provide TSF data management actions in addition to those listed above, these additional actions must be similarly restricted to the authorized administrator.

5.5.7 Management of TSF Data (authentication data)

FMT_MTD.1.1 The TSF shall restrict the ability to initialize the authentication data to authorized administrators and users authorized by the Discretionary Access Control Policy to modify (their own) authentication data.

Application Note: This component applies only to security attributes that are used to maintain the TSP.

While the TOE may provide TSF data management actions in addition to those listed above, these additional actions must be similarly restricted to the authorized administrator.

5.5.8 Management of TSF Data (critical security parameters)

FMT_MTD.1.1 The TSF shall restrict the ability to initialize and modify the critical security parameters to cryptographic administrators.

5.5.9 Revocation (of user access)

FMT_REV.1.1 The TSF shall restrict the ability to revoke security attributes associated with the users within the TSC to authorized administrators.

Application Note: The term "revoke security attributes" means "change attributes so that access is revoked".

FMT_REV.1.2 The TSF shall enforce the rules:

a) The immediate revocation of security-relevant authorizations;

b) Any other revocation rules concerning access control, as defined in the ST.

Application Note: The access control policy may include immediate or delayed revocation. The access rights are considered to have been revoked when all subsequent access control decisions made by the TSF use the new access control information. The choice of whether access is controlled on the basis of users, subjects, or objects is an implementation detail not restricted by this PP; however, the ST must clearly state the basis that is used by the TOE. It is not required that every operation make an explicit access control decision, provided a previous access control decision was made to permit that operation. It is sufficient that the developer clearly describes in guidance documentation how revocation is enforced.

Application Note: It is worth noting that the points at which the access checks are made are never explicitly stated; this requirement requires an implicit statement of where the access checks are made.

5.5.10 Revocation (of access to objects)

FMT_REV.1.1 The TSF shall restrict the ability to revoke security attributes associated with the users, subjects, and objects within the TSC to users authorized to modify the security attributes according to the Discretionary Access Control policy.

Application Note: The term "revoke security attributes" means "change attributes so that access is revoked".

FMT_REV.1.2 The TSF shall enforce the rules:

a) The access rights associated with a user, subject, or object shall be enforced when an access check is made;

b) Any other revocation rules concerning access control, as defined in the ST.

5.5.11 Security Attribute Expiration

FMT_SAE.1.1 The TSF shall restrict the capability to specify an expiration time for security attributes for user authentication passwords to the authorized administrator.

FMT_SAE.1.2 For each of these security attributes, the TSF shall be able to lock out the associated user account after the expiration time for the attribute has passed.

5.5.12 Security Management Roles

FMT_SMR.1.1 The TSF shall maintain the roles:

a) authorized administrator;

Application Note: Any user that is authorized to bypass the DAC policy is, by definition, an authorized administrator. The TOE may provide multiple administrator roles (audit administrator, security administrator, etc). The ST must list all of the types of administrators provided by the TOE.

b) cryptographic administrator (i.e., users authorized to perform cryptographic initialization and management functions);

c) authorized users (i.e., users authorized to use some TOE resources);

d) users authorized by the Discretionary Access Control Policy to modify object security attributes (see FDP_ACF.1.3);

e) users authorized to modify their own authentication data; and

f) other roles defined in the ST.

FMT_SMR.1.2 The TSF shall be able to associate users with roles.

Application Note: The TOE need only support the "authorized administrator" and "cryptographic administrator" roles. If the TOE implements additional independent roles, the ST must specify which roles fulfill each requirement.

FMT_SMR.3.1 The TSF shall require an explicit request to assume the following roles:

a) authorized administrator;

b) cryptographic administrator;

c) other roles defined in the ST.

5.6 Protection of the TOE Security Functions

5.6.1 Underlying Abstract Machine Test

FPT_AMT.1.1 The TSF shall run a suite of tests during the initial start-up, periodically during normal operation, or at the request of an authorized administrator to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the TSF.

Application Note: The test suite need only cover aspects of the underlying abstract machine on which the TSF relies to implement required functions, including domain separation.

5.6.2 Internal TOE TSF Data Transfer

FPT_ITT.1.1 The TSF shall protect TSF data from disclosure when it is transmitted between separate parts of the TOE.

Refinement: This protection must be achieved through the use of encryption when the separate parts of the TOE are distributed.

5.6.3 Internal TOE TSF Data Transfer

FPT_ITT.3.1 The TSF shall be able to detect modification and substitution of data for TSF data transmitted between separate parts of the TOE.

Application Note: This detection must be achieved through the use of cryptographic means when the separate parts of the TOE are transferred.

5.6.4 Trusted Recovery

FPT_RCV.1.1 After a failure or service discontinuity, the TSF shall enter a maintenance mode where the ability to return the TOE to a secure state is provided.

Application Note: Recovery from a failure of the cryptographic module must result in the cryptomodule being in a known and secure state such that all critical areas are empty of plaintext/red/secret data and inaccessible to processes, and all security policies are enforced.

5.6.5 Reference Mediation

FPT_RVM.1.1 The TSF shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

5.6.6 Domain Separation

FPT_SEP.1.1 The TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

FPT_SEP.1.2 The TSF shall enforce separation between the distinct security domains of subjects in the TSC.

Application Note: This component does not imply a particular implementation of TOE. It requires that the implementation exhibit properties that the code and data upon which observation of TSF data would not result in a failure of the TSF to perform its job. This could be accomplished either by hardware mechanisms (for example, by multi-state CPUs which support multiple task spaces) or by hardware architecture (for example, independent nodes within a distributed architecture).

The second element can also be satisfied in a variety of ways, including CPU support for separate address spaces, separate hardware components, or entirely in software, in the case of layered application such as a graphic user interface system which maintains separate objects.

5.6.7 Time Stamps

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps for its own use.

Application Note: The generation of audit records depends upon having a correct date and time, The ST needs to specify the degree of accuracy that must be maintained in order to maintain useful information for audit records.

5.6.8 Inter-TSF TSF Data Consistency

FPT_TDC.1.1 The TSF shall provide the capability to consistently interpret objects and their security attributes when shared between the TSF and another trusted IT product.

FPT_TDC.1.2 The TSF shall use the rules specified in the ST when interpreting the TSF data from another trusted IT product.

5.6.9 Internal TOE TSF Data Replication Consistency

FPT_TRC.1.1 The TSF shall ensure that TSF data is consistent when replicated between parts of the TOE.

FPT_TRC.1.2 When parts of the TOE containing replicated TSF data are disconnected, the TSF shall ensure the consistency of the replicated TSF data upon reconnection before processing any requests for access to objects by users.

5.6.10 TSF Testing

FPT_TST.1.1 The TSF shall run a suite of self tests during the initial start-up, periodically during normal operation, or at the request of an authorized administrator to demonstrate the correct operation of the TSF.

FPT_TST.1.2 The TSF shall provide authorized users with the capability to verify the integrity of TSF data.

Refinement: The term "authorized users" refers to the "authorized administrators" identified in FMT_SMR.1.

FPT_TST.1.3 The TSF shall provide authorized users with the capability to verify the integrity of stored TSF executable code.

Refinement: The term "authorized users" refers to the "authorized administrators" identified in FMT_SMR.1.

5.6.11 Extension: Cryptographic Module Testing

EXTENDED_FPT_CTST.1.1: To demonstrate the correct operation of the cryptographic module, the TSF shall run a suite of self-tests upon initial start-up, at the request of the cryptographic administrator, and periodically to verify the correct operation of the critical security functions, as identified in the ST. The design of the TSF shall not preclude self-tests at least once per day.

Refinement: These self-tests shall be in accordance with FIPS PUB 140-1, Level 4 (as interpreted in the following table) and with any augmented requirements identified in this PP.

	Sec. Level 1	**Sec. Level 2**	**Sec. Level 3**	**Sec. Level 4**
Software/Firmware Integrity Tests	on power on on demand	on power on on demand	on power on on demand	**on power on** **on demand** **conditional**
Cryptographic Algorithm Tests	on power on on demand	on power on on demand	on power on on demand	**on power on** **on demand** **conditional**
Other Critical functions tests as determined by FIPS PUB 140-1, Appendix A	on power on on demand	on power on on demand	on power on on demand	**on power on** **on demand** **conditional**
Statistical RNG/PRNG tests	No Requirement	No Requirement	on demand only	**on power on** **on demand**

Table 5.2 - Interpretation of FIPS PUB 140-1 Self-tests

EXTENDED_FPT_CTST.1.2: Tests must be performed and documented to demonstrate that the RNG/PRNGs do NOT have start-up patterns that are biased or predictable.

Application Note: For non-deterministic RNGs, this might be done by demonstrating in a prototype, by built-in tests, by some external "proof of concept", or by some other means.

EXTENDED_FPT_CTST.1.3: (Conditional Test) The TSF shall perform a key error detection check on each transfer of key (internal, intermediate transfers)

EXTENDED_FPT_CTST.1.4: (Conditional Test) In addition to the prescribed FIPS PUB 140-1, Level 4 self-tests and augmented tests in this PP, the TSF shall perform self-test of each key generation component immediately after generation of a key to verify that each of these components operates in accordance with an RNG/PRNG as specified in this PP. The key generated shall not be used if any one of the component self-tests fails.

Application Note: Key generation components are those critical elements that compose the entire key generation process (e.g., any algorithms, any RNG/PRNGs, any key generation seeding processes, etc.).

5.7 Resource Utilization

5.7.1 Extension: Resource Allocation

FRU_RSA.1.1 The TSF shall enforce maximum quotas of the following resources: percentage of disk space and percentage of system memory that individual users can use at any given time.

5.7.2 Extension: Resource Allocation

FRU_RSA.1.1 The TSF shall enforce maximum quotas of the following resources: percentage of processing time that subjects can use at any given time.

5.8 TOE access

5.8.1 Session Locking

FTA_SSL.1.1 The TSF shall lock an interactive session after a specified time interval of user inactivity by:

a) Clearing or overwriting display devices, making the current contents unreadable.

b) Disabling any activity of the user's data access/display devices other than unlocking the session.

c) Other means of locking the interactive session, as defined in the ST.

FTA_SSL.1.2 The TSF shall require the following event to occur prior to unlocking the session:

a) The TSF shall require the user to re-authenticate prior to unlocking the session (see FIA_AFL.1.2 and FTP_TRP.1).

b) Other events, as defined in the ST.

FTA_SSL.2.1 The TSF shall allow user-initiated locking of the user's own interactive session by:

a) Clearing or overwriting display devices, making the current contents unreadable.

b) Disabling any activity of the user's data access/display devices other than unlocking the session.

c) Other means of locking the interactive session, as defined in the ST.

FTA_SSL.2.2 The TSF shall require the following event to occur prior to unlocking the session:

a) The TSF shall require the user to re-authenticate prior to unlocking the session.(see FIA_AFL.1.2).

5.8.2 TOE Access Banners

FTA_TAB.1.1 Before establishing a user session, the TSF shall display an advisory notice and consent warning message regarding unauthorized use of the TOE.

5.8.3 TOE Access History

FTA_TAH.1.1 Upon successful session establishment, the TSF shall display the date, time, and location of the last successful session establishment to the user.

FTA_TAH.1.2 Upon successful session establishment, the TSF shall display the date, time, and location of the last unsuccessful attempt to session establishment and the number of unsuccessful attempts since the last successful session establishment.

FTA_TAH.1.3 The TSF shall not erase the access history information from the user interface without giving the user the opportunity to review the information.

5.9 Trusted path/channel

5.9.1 Trusted Path

FTP_TRP.1.1 The TSF shall provide a communication path between itself and remote and local users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from modification or disclosure.

Application Note: This “distinct” path is merely invoked for the duration of its being needed (e.g., for reauthenticating the user); it need not be invoked for the duration of the user’s session.

FTP_TRP.1.2 The TSF shall permit local users and remote users to initiate communication via the trusted path.

FTP_TRP.1.3 The TSF shall require the use of the trusted path for initial user authentication.

5.9.2 Extension: Trusted Path

EXTENDED_FTP_TRP.1.1 The cryptographic module shall provide a communication path between itself and local users that is logically distinct from other communication paths and provides assured identification of itself.

Application Note: This “distinct” path is merely invoked for the duration of its being needed (e.g., for reauthenticating the user); it need not be invoked for the duration of the user’s session.

6. Security Assurance Requirements

This section contains the detailed security assurance requirements for operating systems supporting single-level system high systems (not to exceed Secret) in environments requiring medium robustness. These security assurance requirements are selected from Part 3 of the Common Criteria.

6.1 Configuration Management

6.1.1 CM Automation

ACM_AUT.1.1D The developer shall use a CM system.

ACM_AUT.1.2D The developer shall provide a CM plan.

ACM_AUT.1.1C The CM system shall provide an automated means by which only authorized changes are made to the TOE implementation representation.

ACM_AUT.1.2C The CM system shall provide an automated means to support the generation of the TOE.

ACM_AUT.1.3C The CM plan shall describe the automated tools used in the CM system.

ACM_AUT.1.4C The CM plan shall describe how the automated tools are used in the CM system.

ACM_AUT.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

6.1.2 CM Capabilities

ACM_CAP.3.1D The developer shall provide a reference for the TOE.

ACM_CAP.3.2D The developer shall use a CM system.

ACM_CAP.3.3D The developer shall provide CM documentation.

ACM_CAP.3.1C The reference for the TOE shall be unique to each version of the TOE.

ACM_CAP.3.2C The TOE shall be labeled with its reference.

ACM_CAP.3.3C The CM documentation shall include a configuration list and a CM plan.

ACM_CAP.3.4C The configuration list shall describe the configuration items that comprise the TOE.

ACM_CAP.3.5C The CM documentation shall describe the method used to uniquely identify the configuration items.

ACM_CAP.3.6C The CM system shall uniquely identify all configuration items.

ACM_CAP.3.7C The CM plan shall describe how the CM system is used.

ACM_CAP.3.8C The evidence shall demonstrate that the CM system is operating in accordance with the CM plan.

ACM_CAP.3.9C The CM documentation shall provide evidence that all configuration items have been and are being effectively maintained under the CM system.

ACM_CAP.3.10C The CM system shall provide measures such that only authorized changes are made to the configuration items.

ACM_CAP.3.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

6.1.3 CM Scope

ACM_SCP.2.1D The developer shall provide CM documentation

ACM_SCP.2.1C The CM documentation shall show that the CM system, as a minimum, tracks the following: the TOE implementation representation, design documentation, test documentation, user documentation, administrator documentation, CM documentation, and security flaws.

ACM_SCP.2.2C The CM documentation shall describe how configuration items are tracked by the CM system.

ACM_SCP.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

6.2 Delivery and Operation

6.2.1 Delivery

ADO_DEL.2.1D The developer shall document procedures for delivery of the TOE or parts of it to the user.

ADO_DEL.2.2D The developer shall use the delivery procedures.

ADO_DEL.2.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user’s site.

ADO_DEL.2.2C The delivery documentation shall describe how the various procedures and technical measures provide for the detection of modifications, or any discrepancy between the developer’s master copy and the version received at the user site.

ADO_DEL.2.3C The delivery documentation shall describe how the various procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user’s site.

ADO_DEL.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

6.2.2 Installation, Generation and Start-up

ADO_IGS.1.1D The developer shall document procedures necessary for the secure installation, generation, and start-up of the TOE.

ADO_IGS.1.1C The documentation shall describe the steps necessary for secure installation, generation, and start-up of the TOE.

ADO_IGS.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADO_IGS.1.2E The evaluator shall determine that the installation, generation, and start-up procedures result in a secure configuration.

6.3 Development Documentation

6.3.1 Functional Specification

ADV_FSP.2.1D The developer shall provide a functional specification

ADV_FSP.2.1C The functional specification shall describe the TSF and its external interfaces using an informal style.

ADV_FSP.2.2C The functional specification shall be internally consistent.

ADV_FSP.2.3C The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing complete details of all effects, exceptions and error messages.

ADV_FSP.2.4C The functional specification shall completely represent the TSF.

ADV_FSP.2.5C The functional specification shall include rationale that the TSF is completely represented.

ADV_FSP.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.2.2E The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

6.3.2 High-Level Design

ADV_HLD.2.1D The developer shall provide the high-level design of the TSF.

ADV_HLD.2.1C The presentation of the high-level design shall be informal.

ADV_HLD.2.2C The high-level design shall be internally consistent.

ADV_HLD.2.3C The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV_HLD.2.4C The high-level design shall describe the security functionality provided by each subsystem of the TSF.

ADV_HLD.2.5C The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.

ADV_HLD.2.6C The high-level design shall identify all interfaces to the subsystems of the TSF.

ADV_HLD.2.7C The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.

ADV_HLD.2.8C The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV_HLD.2.9C The high-level design shall describe the separation of the TOE into TSP-enforcing and other subsystems.

Refinement: The high-level design must identify the cryptographic boundary.

ADV_HLD.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_HLD.2.2E The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

6.3.3 Implementation Representation

ADV_IMP.2.1D The developer shall provide the implementation representation for the entire TSF.

ADV_IMP.2.1C The implementation representation shall unambiguously define the TSF to a level of detail such that the TSF can be generated without further design decisions.

ADV_IMP.2.2C The implementation representation shall be internally consistent.

ADV_IMP.2.3C The implementation representation shall describe the relationships between all portions of the implementation.

ADV_IMP.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_IMP.2.2E The evaluator shall determine that the least abstract TSF representation provided is an accurate and complete instantiation of the TOE security functional requirements.

6.3.4 TSF Internals

ADV_INT.1.1D The developer shall design and structure the TSF in a modular fashion that avoids unnecessary interactions between the modules of the design.

Application Note: The processing space used by the cryptographic module must be separated from that used by other trusted processes. There must be a means provided to callers of the cryptographic module so they can determine they are really communicating with it (see FTP_TRP.1). The architecture must be such that the inputs and outputs to/from the cryptographic module is distinct and separate.

ADV_INT.1.2D The developer shall provide an architectural description.

ADV_INT.1.1C The architectural description shall identify the modules of the TSF.

ADV_INT.1.2C The architectural description shall describe the purpose, interface, parameters, and effects of each module of the TSF.

ADV_INT.1.3C The architectural description shall describe how the TSF design provides for largely independent modules that avoid unnecessary interactions.

ADV_INT.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_INT.1.2E The evaluator shall determine that both the low-level design and the implementation representation are in compliance with the architectural description.

6.3.5 Low-Level Design

ADV_LLD.1.1D The developer shall provide the low-level design of the TSF.

ADV_LLD.1.1C The presentation of the low-level design shall be informal.

ADV_LLD.1.2C The low-level design shall be internally consistent.

ADV_LLD.1.3C The low-level design shall describe the TSF in terms of modules.

ADV_LLD.1.4C The low-level design shall describe the purpose of each module.

ADV_LLD.1.5C The low-level design shall define the interrelationships between the modules in terms of provided security functionality and dependencies on other modules.

Refinement: The low-level design must include a description or diagram of the state-transitions of the cryptographic module.

ADV_LLD.1.6C The low-level design shall describe how each TSP-enforcing function is provided.

ADV_LLD.1.7C The low-level design shall identify all interfaces to the modules of the TSF.

Refinement: The physical/logical ports of the cryptographic module must be identified.

ADV_LLD.1.8C The low-level design shall identify which of the interfaces to the modules of the TSF are externally visible.

ADV_LLD.1.9C The low-level design shall describe the purpose and method of use of all interfaces to the modules of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV_LLD.1.10C The low-level design shall describe the separation of the TOE into TSP-enforcing and other modules.

Refinement: The cryptographic boundary must be identified in the low-level design.

ADV_LLD.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_LLD.1.2E The evaluator shall determine that the low-level design is an accurate and complete instantiation of the TOE security functional requirements.

6.3.6 Representation Correspondence

ADV_RCR.1.1D The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided.

ADV_RCR.1.1C For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

ADV_RCR.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

6.3.7 Security Policy Modeling

ADV_SPM.1.1D The developer shall provide a TSP model.

ADV_SPM.1.2D The developer shall demonstrate correspondence between the functional specification and the TSP model.

ADV_SPM.1.1C The TSP model shall be informal.

ADV_SPM.1.2C The TSP model shall describe the rules and characteristics of all policies of the TSP that can be modeled.

Application Note: The Informal Model must include descriptions of at least the following security policies: Identification and Authentication, Discretionary Access Control, Audit, and Cryptography.

ADV_SPM.1.3C The TSP model shall include a rationale that demonstrates that it is consistent and complete with respect to all policies of the TSP that can be modeled.

ADV_SPM.1.4C The demonstration of correspondence between the TSP model and the functional specification shall show that all of the security functions in the functional specification are consistent and complete with respect to the TSP model.

ADV_SPM.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

6.4 Guidance Documents

6.4.1 Administrator Guidance

AGD_ADM.1.1D The developer shall provide administrator guidance addressed to system administrative personnel.

AGD_ADM.1.1C The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.

Refinement: The TOE must support at least the "authorized administrator" and "cryptographic administrator" roles (see FMT_SMR.1).

AGD_ADM.1.2C The administrator guidance shall describe how to administer the TOE in a secure manner.

AGD_ADM.1.3C The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

AGD_ADM.1.4C The administrator guidance shall describe all assumptions regarding user behavior that are relevant to secure operation of the TOE.

AGD_ADM.1.5C The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.

AGD_ADM.1.6C The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_ADM.1.7C The administrator guidance shall be consistent with all other documentation supplied for evaluation.

AGD_ADM.1.8C The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.

ADV_ADM.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

6.4.2 User Guidance

AGD_USR.1.1D The developer shall provide user guidance.

AGD_USR.1.1C The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.

Application Note: This includes guidance for the users of the cryptographic module.

AGD_USR.1.2C The user guidance shall describe the use of user-accessible security functions provided by the TOE.

AGD_USR.1.3C The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.

AGD_USR.1.4C The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behavior found in the statement of TOE security environment.

AGD_USR.1.5C The user guidance shall be consistent with all other documentation supplied for evaluation.

AGD_USR.1.6C The user guidance shall describe all security requirements for the IT environment that are relevant to the user.

AGD_USR.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

6.5 Life Cycle Support

6.5.1 Development Security

ALC_DVS.1.1D The developer shall produce development security documentation.

ALC_DVS.1.1C The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

ALC_DVS.1.2C The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.

ALC_DVS.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_DVS.1.2E The evaluator shall confirm that the security measures are being applied.

6.5.2 Flaw Remediation

ALC_FLR.1.1D The developer shall document the flaw remediation procedures.

ALC_FLR.1.1C The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

ALC_FLR.1.2C The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC_FLR.1.3C The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC_FLR.1.4C The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.

ALC_FLR.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

6.5.3 Life Cycle Definition

ALC_LCD.1.1D The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.

ALC_LCD.1.2D The developer shall provide life-cycle definition documentation.

ALC_LCD.1.1C The life-cycle definition documentation shall describe the model used to develop and maintain the TOE.

ALC_LCD.1.2C The life-cycle model shall provide for the necessary control over the development and maintenance of the TOE.

ALC_LCD.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

6.5.4 Tools and Techniques

ALC_TAT.1.1D The developer shall identify the development tools being used for the TOE.

ALC_TAT.1.2D The developer shall document the selected implementation-dependent options of the development tools.

ALC_TAT.1.1C All development tools used for implementation shall be well defined.

Refinement: The compiler used to generate the TOE must be identified.

ALC_TAT.1.2C The documentation of the development tools shall unambiguously define the meaning of all statements used in the implementation.

ALC_TAT.1.3C The documentation of the development tools shall unambiguously define the meaning of all implementation-dependent options.

Refinement: The compiler options used during generation of the TOE must be identified.

ALC_TAT.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

6.6 Testing

6.6.1 Coverage

ATE_COV.2.1D The developer shall provide an analysis of the test coverage.

ATE_COV.2.1C The analysis of the test coverage shall demonstrate the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.

ATE_COV.2.2C The analysis of the test coverage shall demonstrate that the correspondence between the TSF as described in the functional specification and the tests identified in the test documentation is complete.

ATE_COV.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

6.6.2 Depth

ATE_DPT.2.1D The developer shall provide the analysis of the depth of testing.

ATE_DPT.2.1C The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design and low-level design.

ATE_DPT.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

6.6.3 Functional Tests

ATE_FUN.1.1D The developer shall test the TSF and document the results.

ATE_FUN.1.2D The developer shall provide test documentation.

ATE_FUN.1.1C The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.

ATE_FUN.1.2C The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.

ATE_FUN.1.3C The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.4C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.5C The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.

ATE_FUN.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

6.6.4 Independent Testing

ATE_IND.2.1D The developer shall provide the TOE for testing.

ATE_IND.2.1C The TOE shall be suitable for testing.

ATE_IND.2.2C The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

ATE_IND.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.1.2E The evaluator shall test a subset of the TSF as appropriate to confirm that the TOE operates as specified.

6.7 Vulnerability Assessment

6.7.1 Extension: Cryptographic Module Covert Channel Analysis

EXTENDED_AVA_CCA.1.1D For the cryptographic module, the developer shall conduct a search for covert channels for the information flow control policy and integrity policy.

Application Note: The remainder of the TOE need not be subjected to a covert channel analysis.

EXTENDED_AVA_CCA.1.2D The developer shall provide covert channel analysis documentation.

EXTENDED_AVA_CCA.1.1C The analysis documentation shall identify covert channels in the cryptographic module and estimate their capacity.

EXTENDED_AVA_CCA.1.2C The analysis documentation shall describe the procedures used for determining the existence of covert channels in the cryptographic module, and the information needed to carry out the covert channel analysis.

EXTENDED_AVA_CCA.1.3C The analysis documentation shall describe all assumptions made during the covert channel analysis.

EXTENDED_AVA_CCA.1.4C The analysis documentation shall describe the method used for estimating channel capacity, based on worst case scenarios.

EXTENDED_AVA_CCA.1.5C The analysis documentation shall describe the worst case exploitation scenario for each identified covert channel.

EXTENDED_AVA_CCA.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

EXTENDED_AVA_CCA.1.2E The evaluator shall confirm that the results of the covert channel analysis show that the cryptographic module meets its functional requirements.

EXTENDED_AVA_CCA.1.3E The evaluator shall selectively validate the covert channel analysis through testing.

6.7.2 Misuse

AVA_MSU.1.1D The developer shall provide guidance documentation.

AVA_MSU.1.1C The guidance documentation shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AVA_MSU.1.2C The guidance documentation shall be complete, clear, consistent and reasonable.

AVA_MSU.1.3C The guidance documentation shall list all assumptions about the intended environment.

AVA_MSU.1.4C The guidance documentation shall list all requirements for external security measures (including external procedural, physical and personnel controls).

AVA_MSU.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_MSU.1.2E The evaluator shall repeat all configuration and installation procedures to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.

AVA_MSU.1.3E The evaluator shall determine that the use of the guidance documentation allows all insecure states to be detected.

6.7.3 Strength of TOE security functions

AVA_SOF.1.1D The developer shall perform a strength of TOE security function analysis for each mechanism identified in the ST as having a strength of TOE security function claim.

AVA_SOF.1.1C For each mechanism with a strength of TOE security function claim the a strength of TOE security function analysis shall show that it meets or exceeds the minimum strength level defined in the PP/ST.

AVA_SOF.1.2C For each mechanism with a specific strength of TOE security function claim the a strength of TOE security function analysis shall show that it meets or exceeds the specific strength of function metric defined in the PP/ST.

AVA_SOF.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_SOF.1.2E The evaluator shall confirm that the strength claims are correct.

6.7.4 Vulnerability Analysis

AVA_VLA.3.1D The developer shall perform and document an analysis of the TOE deliverables searching for ways in which a user can violate the TSP.

AVA_VLA.3.2D The developer shall document the disposition of identified vulnerabilities.

AVA_VLA.3.1C The documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.

AVA_VLA.3.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VLA.3.2E The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure the identified vulnerabilities have been addressed.

AVA_VLA.3.3E The evaluator shall perform an independent vulnerability analysis.

7. Rational

This section provides the rationale for the selection, creation, and use of security objectives and requirements.

7.1 Security Objectives derived from Threats

Each of the identified threats to security is either negated by an assumption, or results in a security objective.

Threat	**Negating Assumptions / Resultant Objectives**
T.AUDIT_CORRUPT	A.PHYSICAL, O.AUDIT_PROTECTION, O.TRAINED_USERS
T.CONFIG_CORRUPT	A.PHYSICAL, O.SELF_PROTECTION, O.TRAINED_USERS
T.EAVESDROP	O.ENCRYPTED_CHANNEL, O.TRAINED_USERS
T.IMPROPER_ADMIN	A.ADMIN, A.MANAGE, O.MANAGE, O.INSTALL, O.TRAINED_ ADMIN
T.MASQUERADE	O.TRAINED_USERS, O.TRUSTED_PATH
T.OBJECTS_NOT_CLEAN	O.RESIDUAL_INFORMATION
T.POOR_DESIGN	O.CONFIG_MGMT, O.PENETRATION_TEST, O.SOUND_DESIGN, O.VULNERABILITY_ANALYSIS
T.POOR_IMPLEMENTATION	O.CONFIG_MGMT, O.PENETRATION_TEST, O.SOUND_ IMPLEMENTATION, O.VULNERABILITY_ANALYSIS
T.POOR_TEST	O.TESTING
T.REPLAY	O.ENCRYPTED_CHANNEL, O.TRUSTED_PATH, O.USER_IDENTIFICATION
T.SPOOF	O.TRUSTED_PATH, O.USER_IDENTIFICATION,
T.SYSACC	O.ACCESS, O.USER_IDENTIFICATION
T.UNAUTH_ACCESS	A.PHYSICAL, O.ACCESS,

	O.SELF_PROTECTION,
T.UNAUTH_MODIFICATION	O.SELF_PROTECTION
T.UNDETECTED_ACTIONS	A.PHYSICAL, O.AUDIT_GENERATION
T.UNSECURE_START	A.ADMIN, O.RECOVERY
T.USER_CORRUPT	O.DISCRETIONARY_ACCESS, O.PROTECT

7.2 Objectives derived from Security Policies

Each of the identified security policies implies a set of security objectives to be met.

Policies	**Objectives enforcing Policies**
P.ACCOUNT	O.AUDIT_GENERATION, O.AUDIT_REVIEW
P.AUTHORIZATION	O.USER_IDENTIFICATION
P.AUTHORIZED_USERS	O.USER_IDENTIFICATION
P.IANDA	O.USER_IDENTIFICATION
P.NEED_TO_KNOW	O.DISCRETIONARY_ACCESS
P.REMOTE_ADMIN_ACCESS	A.ADMIN_ACCESS, O.ENCRYPTED_CHANNEL, O.TRUSTED_PATH, O.USER_IDENTIFICATION
P.ROLES	O.USER_IDENTIFICATION
P.TESTING	O.TESTING
P.TRACE	O.AUDIT_REVIEW
P.TRUSTED_RECOVERY	O.RECOVERY
P.VULNERABILITY_SEARCH	O.VULNERABILITY_ANALYSIS

7.3 Requirements Rationale

Each the security objectives identified in sections 7.1 and 7.2 is met by a set of security requirements.

Objectives from policies/threats	**Requirements meeting objectives**
O.ACCESS	5.1.1 Security Audit Automatic Response 5.1.2 Security Audit Data Generation 5.1.3 Security Audit Analysis 5.3.1 Access Control Policy

	5.3.2 Access Control Functions
	5.3.3 Export to Outside OS Control
	5.3.6 Import From Outside OS Control
	5.3.7 Internal TOE Transfer
	5.3.8 Residual Information Protection
	5.4.1 Authentication Failures
	5.4.2 User Attribute Definition
	5.4.3 Specification of Secrets
	5.4.4 User Authentication
	5.4.5 User Identification
	5.4.6 User-Subject Binding
	5.5.2 Management of Security Attributes
	5.5.5 Security Attribute Expiration
	5.7.1 Resource Allocation
	5.8.1 Session Locking
	5.8.3 TOE Access History
O.AUDIT_GENERATION	5.1.2 Security Audit Data Generation
O.AUDIT_PROTECTION	5.1.6 Event Storage
O.AUDIT_REVIEW	5.1.4 Security Audit Review
	5.1.5 Event Selection
O.CONFIG_MGMT	6.1.1 CM Automation
	6.1.2 CM Capabilities
	6.1.3 CM Scope
	6.5.1 Development Security
	6.5.2 Flaw Remediation
	6.5.3 Life Cycle Definition
	6.5.4 Tools and Techniques
O.DISCRETIONARY_ACCESS	5.3.1 Access Control Policy
	5.3.2 Access Control Functions
O.ENCRYPTED_CHANNEL	5.10.1 Cryptographic Key Management
	5.10.2 Cryptographic Operation
O.INSTALL	6.2.1 Delivery

O.MANAGE	5.5.1 Management of Security Functions Behavior
	5.5.2 Management of Security Attributes
	5.5.3 Management of OS Data
	5.5.4 Revocation
	5.5.5 Security Attribute Expiration
	5.5.6 Security Management Roles
O.PENETRATION_TEST	6.6.4 Independent Testing
	6.7.2 Vulnerability Analysis
O.PROTECT	5.1.6 Security Audit Event Storage
	5.5.3 Management of OS Data
	5.5.4 Revocation
	5.6.1 Underlying Abstract Machine Test
	5.6.2 Internal TOE TSF Data Transfer
	5.6.3 Trusted Recovery
	5.6.4 Reference Mediation
	5.6.5 Domain Separation
	5.6.6 Time Stamps
	5.6.7 Internal TSF Data Replication Consistency
	5.6.8 OS Testing
O.RECOVERY	5.6.3 Trusted Recovery
O.RESIDUAL_INFORMATION	5.3.8 Residual Information Protection
O.SELF_PROTECTION	5.6.3 Domain Separation
O.SOUND_DESIGN	6.3.1 Functional Specification
	6.3.2 High-Level Design
	6.3.5 Low-Level Design
	6.3.6 Representation Correspondence
	6.3.7 Security Policy Modeling
O.SOUND_IMPLEMENTATION	6.3.3 Implementation Representation
	6.3.4 TSF Internals
	6.3.6 Representation Correspondence
O.TESTING	6.6.1 Coverage
	6.6.2 Depth

	6.6.3 Functional Tests
	6.6.4 Independent Testing
	6.7.2 Vulnerability Analysis
O.TRAINED_ ADMIN	6.2.2 Installation, Generation and Start-up
	6.4.1 Administrator Guidance
O.TRAINED_USERS	6.4.2 User Guidance
O.TRUSTED_PATH	5.9.1 Trusted Path
	5.10.1 Cryptographic Key Management
	5.10.2 Cryptographic Operation
O.USER_IDENTIFICATION	5.4.1 Authentication Failures
	5.4.2 User Attribute Definition
	5.4.3 Specification of Secrets
	5.4.4 User Authentication
	5.4.5 User Identification
	5.4.6 User-Subject Binding
O.VULNERABILITY_ANALYSIS	6.7.2 Vulnerability Analysis

Appendix A — Acronyms

CC	Common Criteria
COTS	Commercial Off-The-Shelf
CSP	Critical Security Parameters
DAC	Discretionary Access Control
DoD	Department of Defense
EAL	Evaluation Assurance Level
FIPS	Federal Information Processing Standard
GiG	Guidance and Policy for Department of Defense Information Assurance Memorandum No. 6-8510
IT	Information Technology
PP	Protection Profile
SF	Security Function
SFP	Security Function Policy
SOF	Strength of Function
ST	Security Target
TOE	Target of Evaluation
TSC	TSF Scope of Control
TSF	TOE Security Functions
TSFI	TSF Interface
TSP	TOE Security Policy

References

[1] Common Criteria Implementation Board, Common Criteria for Information Technology Security Evaluation, CCIB-98-026, Version 2.0, May 1998

[2] Department of Defense Chief Information Officer, Guidance and Policy for Department of Defense Information Assurance Memorandum No. 6-8510 (Draft) dated 19 January 2000.

[3] National Institute of Standards and Technology, Security Requirements for Cryptographic Modules, Federal Information Processing Standard Publication (FIPS-PUB) 140-1, dated January 11, 1994.

[4] National Security Agency, Labeled Security Protection Profile (LSPP) Version 1.b, 8 October 1999

[5] National Security Agency, Controlled Access Protection Profile (CAPP), Version 1.d, 8 October 1999

[6] National Security Agency, Information Assurance Technical Framework (IATF), Version 2.0.1 - September 1999

[7] Department of Defense Standard, Department of Defense Trusted Computer System Evaluation Criteria (Orange Book), December 1985